Docket No.: MP0274

APPLICATION

FOR

UNITED STATES LETTERS PATENT

TITLE: ALIGNING IP PAYLOADS ON MEMORY BOUNDARIES FOR IMPROVED PERFORMANCE AT A SWITCH

APPLICANT: NAFEA BISHARA

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EV 348 189 365 US

August 26, 2003

Date of Deposit

ALIGNING IP PAYLOADS ON MEMORY BOUNDARIES FOR IMPROVED PERFORMANCE AT A SWITCH

BACKGROUND

[0001] TCP/IP (Transmission Control Protocol/Internet Protocol) is the basic communication protocol of the Internet and may also be used as a communications protocol in private networks (e.g., intranets). TCP/IP is frequently referred to as a protocol stack, which refers to the layers through which data passes at both client and server ends of a data exchange. The TCP/IP model has four layers: an application layer (e.g., FTP, SMTP, or Telnet); a transport layer (TCP or UDP); a network layer (IP); and a link layer (e.g., Ethernet).

[0002] When TCP/IP is implemented in an Ethernet network, Ethernet frames encapsulate the data for the upper layers. Figure 1 illustrates a format for an Ethernet frame 100. The Ethernet frame includes header information 105, a payload 110, and trailer information 115. The header information includes a 6 byte MAC (Media Access Control) destination address (DA) portion 120, a 6 byte MAC source address (SA) portion 125, and a 2 byte Type portion 130. The trailer information 115 includes a 4 byte checksum portion 135. The payload 110 includes an IP

packet with its own header 135 and trailer information 140, and its own payload 145. Network devices in the network may include TCP/IP software stacks, which enable the network device to extract data from the various packets (e.g., Ethernet frames, IP packets, and TCP datagrams) and to encapsulate and format data for transmission over the appropriate layer.

SUMMARY

[0003] A network device, e.g., a switch or a router, in a networked computer system may receive frames and store the frames in a memory having memory regions. The frames include header and payload portions. The header portions of the frames may be of a size that causes the payloads to be misaligned in the memory regions, i.e., not aligned on memory region boundaries, when the frame is stored in the memory.

[0004] The network device may include an alignment module that inserts a dummy portion in the header, either before or after the header. The presence of the dummy portion may shift the position of the payload in the memory such that the payload is aligned on a memory region boundary.

[0005] In an embodiment, the frame may be Ethernet frames, which encapsulate IP packets. The frames may have an n-bit header and the memory may have m-bit memory regions, where the ratio n/m has a non-zero remainder p. The alignment module may append a p-bit dummy region to the header to shift the payload in the memory region such that the payload is aligned on a memory region boundary.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] Figure 1 is a block diagram of an Ethernet frame format.

[0007] Figure 2 is a block diagram of a system including a network device according to an embodiment.

[0008] Figure 3 is a block diagram of memory device including a frame with a misaligned payload.

[0009] Figure 4 is a flowchart describing an alignment operation.

[0010] Figure 5 is a block diagram of an Ethernet frame with prefixed dummy bytes.

[0011] Figure 6 is a block diagram of memory device including a frame with an aligned payload.

[0012] Figure 7 is a block diagram of an Ethernet frame with dummy bytes inserted between the header and the payload.

DETAILED DESCRIPTION

[0013] Figure 2 shows a system according to an embodiment. The system may be part of a networked computer system, e.g., a wired or wireless Ethernet LAN (Local Area Network). The system may include a transceiver 210 and a network device 212. The network device may include a switch 215, a CPU (Central Processing Unit) 220 with a buffer memory 240, and a TCP/IP software stack 235. The CPU 220 may be tightly coupled to the switch. For example, the CPU may be embedded in the switch.

[0014] The transceiver may transmit frames, such as the Ethernet frame 100 shown in Figure 1, to the switch. The switch may channel incoming frames from any of multiple input ports to the specific output port that will take the frame toward its intended destination. The switch may use the physical device (MAC) address in incoming frames to determine which port to forward the frame to and out of. The frame may then be forwarded to another device in the network.

[0015] In some instances, information in the IP packet in the payload of a frame may be needed by the switch. For example, the IP packet may be destined for another network 230. The switch, which may have routing capabilities

(e.g., a Level 3 switch) or include a router, may use information in the IP header to route the IP packet to the other network. The CPU may use the TCP/IP stack 235 to extract the payload (i.e., the IP packet) from the frame. The switch may then use the information in the IP packet header to route the packet.

[0016] The CPU 220 may store a received Ethernet frame in the memory 240. In an embodiment, the memory may be partitioned into 4 byte (32-bit or word) memory regions 245. As shown in Figure 3, in a standard Ethernet frame such as that shown in Figure 1, the payload (e.g., IP packet) may not be aligned on a 4 byte boundary 305 if stored in the memory 240. The 14 bytes of header information including the MAC DA, MAC SA, and Type portions causes the payload to be misaligned by 2 bytes in a memory region 310.

[0017] The operating system utilized by the CPU may require the payload to be aligned on the 4-byte boundaries in the memory 240 for processing. If the payload is misaligned, the TCP/IP stack 235 may copy the payload (e.g., IP packet) into an alternative memory on the 4 byte boundaries (i.e., align the payload) and then use the copy in the alternative memory. However, this extra step may decrease performance in the switch.

[0018] In an embodiment, the network device may include an alignment module 260 that modifies the frames received at the switch to avoid such performance issues. Figure 4 is a flowchart describing an alignment operation according to an embodiment. The alignment module 260 intercepts frames sent to the CPU (block 405) and prefixes two dummy bytes 505 to the beginning of the frames (block 410), as shown in Figure 5. The frame is then stored in the memory (block 415). The dummy bytes extend the header to 16 bytes, which shifts the frame in the memory to align the payload on the 4-byte boundaries 305. The CPU 220 and TCP/IP stack 235 may be configured to ignore the dummy bytes 505, and locate and access the frame header information in the shifted byte locations (block 420). Consequently, the TCP/IP stack does not need to copy the payload to an alternative memory, thereby avoiding the performance issue.

[0019] In an alternative embodiment, the alignment module may suffix the dummy portion to the header, i.e., insert the dummy portion between the header and the payload, as shown in Figure 7.

[0020] Figure 1 illustrates an Ethernet frame in accordance with the IEEE 802.3 standard. However, other types of frames may benefit from the alignment technique.

For example, an Ethernet V2 frame with an 802.1Q (VLAN) tag has an 18 byte header. Appending a two byte dummy portion to the header of such a frame would align the payload on the memory region boundaries 305 of the memory 240 (Figure 3). Other frame formats include, for example, Ethernet V2 (14 byte header), Ethernet with 802.3 LLC/SNAP (22 byte header), and Ethernet with 802.3 LLC/SNAP and 802.1Q tag (26 byte header).

[0021] A number of embodiments have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the invention. For example, blocks in the flowcharts may be skipped or performed out of order and still produce desirable results. Accordingly, other embodiments are within the scope of the following claims.